Diffusions with position dependent volatility and the Metropolis-adjusted Langevin algorithm

T. Xifara[1,3,**], C. Sherlock[1,*], S. Livingstone[2,*], S. Byrne[2,*], M. Girolami[2,*]

Abstract

We propose a new position dependent Metropolis-adjusted Langevin algorithm (MALA) which is based on a Langevin diffusion with invariant density π with respect to Lebesgue measure in contrast with a recently proposed different position dependent MALA (MMALA). We compare analytically the two algorithms, and show that they are equivalent in some cases but are distinct in general. A different specification of the MMALA scheme, based on a Langevin diffusion with invariant density with respect to Hausdorff measure, is also introduced and compared to the other two MALA schemes. A simulation study illustrates the gain in efficiency provided by the new position dependent MALA algorithm.

Keywords: Diffusions, Markov Chain Monte Carlo, Metropolis-adjusted Langevin Algorithm, Riemannian Manifolds

1. Introduction

The Metropolis-adjusted Langevin algorithm (MALA) [*e.g.* 1] and its manifold variant (MMALA) [2] are Markov chain Monte Carlo methods based on diffusions. While theoretical properties of the former are better understood [*e.g.* 1], the latter has been shown to be more effective in practice, producing more efficient estimates for the same computational budget in many experiments [2]. In this article we highlight two properties of the diffusion on which MMALA is based. First, we point out an unfortunate transcription error which has propagated through the literature, whereby a factor of a 1/2 has been missed from one of the terms [3, 2]. Second, we show that the *corrected* diffusion does not have the intended invariant density with respect to Lebesgue measure. It would seem logical that a similar diffusion which *does* preserve the intended probability density may prove a better basis for a Metropolis–Hastings algorithm. We therefore describe such a diffusion and the resulting sampling method, which we call PMALA (position-dependent MALA). We show that the incorrectly transcribed diffusion and that on which PMALA is based are equivalent in some cases, although the former leads to a more computationally costly algorithm; this equivalence explains to some extent why the error has been missed previously. Finally we describe simulation studies based on those in [2] comparing PMALA, MMALA and the corrected MMALA. In terms of effective sample size (ESS) PMALA outperforms the corrected MMALA, and it outperforms MMALA when the two are not equivalent. PMALA outperforms both of the MMALA algorithms in terms of effective sample size (ESS) per second, since even when MMALA and PMALA are equivalent, each step of PMALA involves fewer CPU operations.

*Corresponding author
**Principal corresponding author
Email addresses: xifara@soe.ucsc.edu (T. Xifara), c.sherlock@lancaster.ac.uk (C. Sherlock), samuel.livingstone@ucl.ac.uk (S. Livingstone)
[1]Department of Mathematics and Statistics, Lancaster University.
[2]Department of Statistical Science, University College London.
[3]Department of Applied Mathematics and Statistics, University of California Santa Cruz.

2. Langevin diffusions

A d-dimensional diffusion is a continuous time stochastic process $\boldsymbol{X} = (\boldsymbol{X}_t)_{t\geq 0}$ with almost surely continuous sample paths. It can be (formally) written as a solution to a stochastic differential equation $d\boldsymbol{X}_t = \boldsymbol{b}(\boldsymbol{X}_t)dt + \boldsymbol{\sigma}(\boldsymbol{X}_t)d\boldsymbol{W}_t$ for drift vector $\boldsymbol{b}(\boldsymbol{x})$ and volatility matrix $\boldsymbol{\sigma}(\boldsymbol{x})$, and where $\boldsymbol{W} = (\boldsymbol{W}_t)_{t\geq 0}$ is a standard d-dimensional Wiener process [*e.g.* 4]. Given an initial condition $\boldsymbol{X}_0 = \boldsymbol{x}_0$, a realisation can be approximately simulated using numerical techniques. The Euler–Maruyama method [*e.g.* 5] is among the simplest: for a chosen step-size h a realisation the sequence of random variables $\boldsymbol{X}_h, \boldsymbol{X}_{2h}, ..., \boldsymbol{X}_{nh}$ is approximated using the procedure

$$\boldsymbol{x}_{(i+1)h} = \boldsymbol{x}_{ih} + h\boldsymbol{b}(\boldsymbol{x}_{ih}) + \boldsymbol{\sigma}(\boldsymbol{x}_{ih})\boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \sim \mathrm{N}_d(0, hI_{d\times d})$.

The distribution of the diffusion is described by the Fokker–Planck equation [*e.g.* 6], which relates the evolution of the density $u(\boldsymbol{x}, t)$ for $\boldsymbol{X}_t$ to the drift and volatility $\boldsymbol{b}, \boldsymbol{\sigma}$,

$$\frac{\partial}{\partial t}u(\boldsymbol{x}, t) = -\sum_i \frac{\partial}{\partial x_i}[b_i(\boldsymbol{x})u(\boldsymbol{x}, t)] + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}[V_{ij}(\boldsymbol{x})u(\boldsymbol{x}, t)], \tag{1}$$

where $V(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{x})\boldsymbol{\sigma}(\boldsymbol{x})^T$. If $u(\boldsymbol{x}, t) = \pi(\boldsymbol{x})$ for all t, then the process is *stationary*, and $\pi(\cdot)$ is the *invariant* or *stationary distribution* of the diffusion[4], meaning that if $\boldsymbol{X}_t \sim \pi(\cdot)$ then $\boldsymbol{X}_{t+\tau} \sim \pi(\cdot)$ for all $\tau > 0$ [*e.g.* 6]. One such is the Langevin diffusion, the solution to:

$$d\boldsymbol{X}_t = \frac{1}{2}\nabla \log \pi(\boldsymbol{X}_t)dt + d\boldsymbol{W}_t,\ \boldsymbol{X}_0 = \boldsymbol{x}_0. \tag{2}$$

Setting $\boldsymbol{b}$ and $\boldsymbol{\sigma}$ as in (2) and $u(\boldsymbol{x}, 0) = \pi(\boldsymbol{x})$ gives $\partial u/\partial t = 0$, meaning that the invariant measure for the Langevin diffusion has associated density $\pi(\boldsymbol{x})$ with respect to Lebesgue measure on $\mathbb{R}^d$. Under certain conditions, the Langevin diffusion converges to π at an exponential rate from any starting point [7].

Markov chain Monte Carlo (MCMC) methods involve simulating from a Markov chain which has a desired invariant density $\pi(\boldsymbol{x})$. Expectations from this distribution can be approximated by averaging values across the chain [*e.g.* 8]. A popular MCMC method is the Metropolis–Hastings algorithm, in which at each iteration some proposal $\boldsymbol{x}'$ is drawn from a distribution $q(.|\boldsymbol{x})$ (where $\boldsymbol{x}$ represents the current value in the chain). The next value in the chain is set to be $\boldsymbol{x}'$ with probability $\alpha(\boldsymbol{x}'|\boldsymbol{x})$, or else $\boldsymbol{x}$, where:

$$\alpha(\boldsymbol{x}'|\boldsymbol{x}) = 1 \wedge \frac{\pi(\boldsymbol{x}')q(\boldsymbol{x}|\boldsymbol{x}')}{\pi(\boldsymbol{x})q(\boldsymbol{x}'|\boldsymbol{x})}. \tag{3}$$

Any diffusion can form the basis of a Metropolis–Hastings algorithm with invariant density π, by setting the proposal density as $\boldsymbol{x}' \sim \mathrm{N}_d(\boldsymbol{x} + b(\boldsymbol{x})h, hV(\boldsymbol{x}))$. Since the objective is to simulate a chain with a certain invariant density, π, basing a scheme on a Langevin diffusion (2) which itself has invariant density π would seem logical, and indeed the diffusion (2) is the basis of the MALA method, whereby proposals are generated according to:

$$\boldsymbol{x}' \sim \mathrm{N}_d\left(\boldsymbol{x} + \frac{h}{2}\nabla \log \pi(\boldsymbol{x}), hI_{d\times d}\right),$$

for a chosen step size h, and then accepted with probability $\alpha(\boldsymbol{x}'|\boldsymbol{x})$. Scaling properties of h with d and asymptotic optimal acceptance rates for the method are discussed in [1]. A slight generalisation of (2) is the diffusion:

$$d\boldsymbol{X}_t = \frac{1}{2}A\nabla \log \pi(\boldsymbol{X}_t)dt + \sqrt{A}\ d\boldsymbol{W}_t, \tag{4}$$

[4]Throughout this article π (or $\pi(\boldsymbol{x})$) refers to a density, which unless otherwise specified is with respect to Lebesgue measure, while $\pi(\cdot)$ is a distribution with density π.

where A is some positive-definite matrix, and $\sqrt{A}$ is any matrix S such that $SS^T = A$. As with the diffusion (2), substitution of the drift and volatility terms from (4) into the Fokker–Planck equation leads to $\partial u/\partial t = 0$, so that π is the invariant density of (4). The Metropolis–Hastings scheme derived from (4) is known as 'pre-conditioned MALA' [3] and is well-suited to scenarios in which the components of π are highly correlated or have very different marginal variances, but where these relationships vary little over the main posterior mass.

The MMALA algorithm [2] is based on the discretisation of a diffusion with a position-dependent volatility matrix:

$$d\boldsymbol{X}_t = \frac{1}{2}G^{-1}(\boldsymbol{X}_t)\nabla \log \pi(\boldsymbol{X}_t)dt + \Omega(\boldsymbol{X}_t)dt + \sqrt{G^{-1}(\boldsymbol{X}_t)}\; d\boldsymbol{W}_t, \tag{5}$$
$$\Omega_i(\boldsymbol{X}_t) = |G(\boldsymbol{X}_t)|^{-1/2}\sum_j \frac{\partial}{\partial X_j}\left[G^{-1}_{ij}(\boldsymbol{X}_t)|G(\boldsymbol{X}_t)|^{1/2}\right],$$

where $G(\boldsymbol{X}_t)$ is some positive definite $d\times d$ matrix. The choice of G is arbitrary, but some natural candidates arise by noting that the above process can be thought of as a diffusion defined on a Riemannian manifold, specified in local coordinates ([2]). In the resulting algorithm, proposals are generated according to:

$$\boldsymbol{x}' = \mathrm{N}_d\left(\boldsymbol{x} + \frac{h}{2}G^{-1}(\boldsymbol{x})\nabla \log \pi(\boldsymbol{x}) + h\Omega(\boldsymbol{x}), hG^{-1}(\boldsymbol{x})\right), \tag{6}$$

and then accepted or rejected according to (3). A similar scheme is proposed in [3], based on the same diffusion. For a suitable choice of $G(\boldsymbol{x})$, allowing a position-dependent covariance matrix for proposals as in (6) allows adaptation to the local curvature of the target density π, which has been shown to increase algorithm efficiency in a number of examples [2].

3. A new position-dependent MALA

In general, a diffusion with invariant density π can be constructed by starting from (1) and selecting a drift and volatility such that

$$b_i(\boldsymbol{x})\pi(\boldsymbol{x}) = \frac{1}{2}\sum_j \frac{\partial}{\partial x_j}\left[V_{ij}(\boldsymbol{x})\pi(\boldsymbol{x})\right]. \tag{7}$$

If the intention is to derive a Metropolis–Hastings proposal mechanism with a position-dependent covariance matrix, a natural starting point would be to simply set $A = A(\boldsymbol{X}_t)$ in (4), giving $d\boldsymbol{X}_t = \frac{1}{2}A(\boldsymbol{X}_t)\nabla \log \pi(\boldsymbol{X}_t)dt + \sqrt{A(\boldsymbol{X}_t)}\; d\boldsymbol{W}_t$, and this diffusion forms the basis of the simplified MMALA algorithm of [2]. However, substituting the drift and volatility terms into (7) gives the requirement that:

$$\frac{1}{2}\sum_j A_{ij}(\boldsymbol{x})\frac{\partial}{\partial x_j}\left[\log \pi(\boldsymbol{x})\right]\pi(\boldsymbol{x}) = \frac{1}{2}\sum_j\left(\frac{\partial A_{ij}(\boldsymbol{x})}{\partial x_j}\pi(\boldsymbol{x}) + A_{ij}(\boldsymbol{x})\frac{\partial \pi(\boldsymbol{x})}{\partial x_j}\right) \tag{8}$$

for each i. Noting that $\partial/\partial x_j[\log \pi(\boldsymbol{x})]\pi(\boldsymbol{x}) = \partial\pi(\boldsymbol{x})/\partial x_j$, it is clear that (8) is only satisfied in general when A is a constant matrix. It is also clear, however, that a simple modification to the drift term will result in (7) being satisfied for π, resulting in a new diffusion:

$$d\boldsymbol{X}_t = \frac{1}{2}A(\boldsymbol{X})\nabla \log \pi(\boldsymbol{X}_t)dt + \Gamma(\boldsymbol{X}_t)dt + \sqrt{A(\boldsymbol{X}_t)}\; d\boldsymbol{W}_t \tag{9}$$
$$\Gamma_i(\boldsymbol{X}_t) = \frac{1}{2}\sum_j \frac{\partial}{\partial X_j}A_{ij}(\boldsymbol{x}).$$

This diffusion again clearly has invariant density π with respect to Lebesgue measure, and the additional drift term Γ is of a simpler form than Ω in (5). The resulting Metropolis–Hastings proposal mechanism is:

$\boldsymbol{x}' \sim \mathrm{N}_d\left(\boldsymbol{x} + \frac{h}{2}A(\boldsymbol{x})\nabla \log \pi(\boldsymbol{x}) + h\Gamma(\boldsymbol{x}), hA(\boldsymbol{x})\right)$. We refer to the resulting Metropolis–Hastings method as 'position-dependent MALA', or more succinctly 'PMALA'.

The remainder of this section details two separate connections between the diffusions (5) and (9) when $A(\boldsymbol{X}_t) = G^{-1}(\boldsymbol{X}_t)$. In describing these connections it will be helpful to note equivalent forms for the ith components of $\Omega(\boldsymbol{X}_t)$ and $\Gamma(\boldsymbol{X}_t)$. For clarity of exposition in the following we suppress explicit dependence on X_t of all four of these quantities.

$$\Omega_i = \sum_j \frac{\partial G^{-1}_{ij}}{\partial X_j} + \frac{1}{2}\sum_j G^{-1}_{ij}\frac{\partial \log |G|}{\partial X_j} \tag{10}$$

$$= -\sum_{jkm} G^{-1}_{ik}\frac{\partial G_{km}}{\partial X_j}G^{-1}_{mj} + \frac{1}{2}\sum_{jkm} G^{-1}_{ij}\frac{\partial G_{mk}}{\partial X_j}G^{-1}_{km}. \tag{11}$$

$$\Gamma_i = \frac{1}{2}\sum_j \frac{\partial G^{-1}_{ij}}{\partial X_j} = -\frac{1}{2}\sum_{jkm} G^{-1}_{ik}\frac{\partial G_{km}}{\partial X_j}G^{-1}_{mj} \tag{12}$$

For the first connection we note that the diffusion (5) on which both MMALA and the algorithm of [3] are based contains a transcription error. The term Ω should in fact be multiplied by a factor of $1/2$, giving the diffusion

$$d\boldsymbol{X}_t = \frac{1}{2}G^{-1}(\boldsymbol{X}_t)\nabla \log \pi^*(\boldsymbol{X}_t)dt + \frac{1}{2}\Omega(\boldsymbol{X}_t)dt + \sqrt{G^{-1}(\boldsymbol{X}_t)}\, d\boldsymbol{W}_t. \tag{13}$$

This can be viewed as a deterministic mapping of (2) onto a Riemannian manifold with metric tensor G, with the first term being the covariant drift, and the second and third corresponding to a Brownian motion on the manifold [9]. However, the density π^* is not given with respect to the Lebesgue measure, but instead respect to the d-dimensional volume or Hausdorff measure of the manifold. We refrain from discussing this in detail, but note that this is related to the density π with respect to the Lebesgue measure via the *area formula* [10, Theorem 3.2.5],

$$\pi(\boldsymbol{x}) = \pi^*(\boldsymbol{x})|G(\boldsymbol{x})|^{1/2}. \tag{14}$$

Lemma. *The diffusions defined by (13) and (9) are equal.*

Proof. The volatilities of the two diffusions are the same, so we need only compare the drift terms. Substituting (14) in (9) gives a diffusion where the ith component of the drift term is

$$b_i = \frac{1}{2}\sum_j G^{-1}_{ij}\frac{\partial \log \pi^*}{\partial X_j} + \frac{1}{4}|G|\sum_j G^{-1}_{ij}\frac{\partial |G|}{\partial X_j} + \frac{1}{2}\sum_j \frac{\partial G^{-1}_{ij}}{\partial X_j}$$

which, using (10), is the ith component of the drift in (13). □

Thus the diffusion (5) arises as a result of an error in transcription, as well as using the density of the incorrect reference measure. Interestingly, in certain circumstances these two mistakes appear to cancel, and that (5) will indeed have the correct invariant distribution.

Proposition. *If $G(\boldsymbol{x})$ is chosen such that for any combination of $1 \leq j, k, m \leq d$:*

$$\frac{\partial}{\partial x_j}G_{km}(\boldsymbol{x}) = \frac{\partial}{\partial x_k}G_{jm}(\boldsymbol{x}) \tag{15}$$

for all $\boldsymbol{x}$, then (5) and (9) represent the same diffusion.

Proof. Since the volatitilites and the multipliers of $\nabla \log \pi$ in the drift are identical for the two diffusions, we need only show that $\Omega_i = \Gamma_i$ for all i. From (15) the second term in (11) can be written as

$$\frac{1}{2}\sum_{jkm} G^{-1}_{ij}\frac{\partial G_{jm}}{\partial X_k}G^{-1}_{km} = \frac{1}{2}\sum_{jkm} G^{-1}_{ik}\frac{\partial G_{km}}{\partial X_j}G^{-1}_{jm},$$

on relabelling $j \leftrightarrow k$. The result follows since $G^{-1}_{jm} = G^{-1}_{mj}$. □

We note that this property arises in certain simple cases, which suggests perhaps how this mistake has thus far remained undetected. Certainly, if the process is univariate ($d = 1$), then condition (15) holds trivially. More generally, it also holds if G is the Hessian matrix of some real-valued function: in particular, in the case of a natural exponential family, such as a generalised linear model (GLM) with canonical link, the Fisher information matrix used by [2], is equal to the Hessian of the negative log-likelihood function.

In general, however, the diffusion (5) will not have the desired invariant density. Based on (1), the necessary and sufficient condition for π to be the invariant density is:

$$\sum_{i,j} \left\{ G^{-1}(\boldsymbol{x}) \frac{\partial G(\boldsymbol{x})}{\partial x_j} G^{-1}(\boldsymbol{x}) \right\}_{ij} = \sum_{i,j} G^{-1}_{ij}(\boldsymbol{x}) \mathrm{tr} \left\{ \frac{\partial G(\boldsymbol{x})}{\partial x_j} G^{-1}(\boldsymbol{x}) \right\}. \tag{16}$$

As (16) does not depend on π it is sufficient to exhibit a positive-definite matrix G which does not satisfy (16).

Example. Set

$$G(\boldsymbol{x}) = \begin{pmatrix} 1 + x^2 & xy \\ xy & 1 + y^2 \end{pmatrix}.$$

The left and right-hand sides of (16) are respectively: $\big(x^3 + y^3 + xy^2 + x^2y + 3x + 3y\big)/(1 + x^2 + y^2)^2$ and $(2x + 2y)/(1 + x^2 + y^2)^2$. Since the two are not in general equivalent any diffusion of the form (5) with G as above will not have the intended invariant density; in fact it need not have an invariant density.

4. Experiments

We compared the performance of the MALA schemes across three of the scenarios considered in [2]: logistic regression on each of five different datasets; a stochastic volatility model; a non-linear ODE model. As in [2] we base the metric tensor, $G(\mathbf{x}) = A(\mathbf{x})^{-1}$, on the expected Fisher information.

Initial tuning runs allowed us to obtain the optimal scaling parameters ($\sqrt{h}$ in this article) in terms of ESS for each algorithm (on each dataset, where relevant). The initialisation, burn-in, and length of each Markov chain was exactly as in [2], however we performed 100 (rather than 10) replicated runs for each chain.

Bayesian logistic regression and the non-linear ODE model are of most interest since in [2] MMALA was found to outperform Riemann Manifold Hamiltonian Monte Carlo for these scenarios. Due to space considerations we therefore present detailed results for these scenarios; results for the stochastic volatility model showed the same underlying pattern. Where especially pertinent we provide brief details on the models themselves and the priors; for further details the reader is referred to [2].

4.1. Logistic regression

We perform Bayesian inference for a logistic regression model on each of five different datasets containing between 7 and 25 covariates. We choose a Gaussian prior for the parameter vector $\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \alpha I)$, so that with a design matrix is X and link function $s(\cdot)$ the metric tensor is given by $G(\boldsymbol{\beta}) = X^T \Lambda X + \alpha^{-1} I$, where Λ is a diagonal matrix with elements $\Lambda_{i,i} = s(\boldsymbol{\beta}^T X^T_{i,\cdot})(1 - s(\boldsymbol{\beta}^T X^T_{i,\cdot}))$. This fulfills the conditions for Corollary 2, so the diffusions on which PMALA and MMALA are based have the same law and we should expect the ESSs for these two algorithms to be the same up to Monte Carlo error.

For each Markov chain the ESS was computed for each parameter and the minimum, median and maximum of these was noted. Table 1 shows, for each algorithm and dataset, the means and their corresponding standard errors using the 100 replicates. The CPU time and the mean (over replicates) minimum (over parameters) effective number of independent samples per second are also provided.

As expected, the ESSs for PMALA and MMALA are very similar. Since Γ is computationally less costly to calculate than Ω, PMALA is quicker than the other two algorithms and so obtains the largest ESS per second.

Table 1: Results for the MMALA schemes for Bayesian logistic regression. The mean (over the 100 replicates) and its standard error is presented for the minimum, median and maximum ESSs (over the parameters). The CPU time and the mean minimum ESS per second are also given.

Dataset	*Method*	*ESS (mean)*	*ESS (s.e)*	*CPU Time*	*mean min. ESS/s*
	PMALA	(685, 847, 986)	(5.5, 3, 4.1)	12.58	54.5
Australian Credit	MMALA	(696, 848, 943)	(6, 2.9, 4.1)	14.08	49.4
	PMALA	(605, 777, 917)	(5.4, 2.5, 4)	43.8	13.8
German Credit	MMALA	(605, 774, 921)	(5.5, 2.5, 3.9)	45.72	13.2
	PMALA	(659, 795, 923)	(5.4, 3.3, 4.3)	6.57	100.3
Heart	MMALA	(657, 773, 920)	(4.8, 2.9, 4.7)	8.07	81.4
	PMALA	(1235, 1415, 1572)	(8.7, 5.9, 6.6)	4.67	264.5
Pima Indian	MMALA	(1264, 1425, 1576)	(9.6, 6.5, 7.6)	5.59	226.1
	PMALA	(477, 591, 679)	(6.8, 5.1, 5)	3.32	143.7
Ripley	MMALA	(460, 590, 686)	(7.5, 5.2, 5.3)	3.94	116.7

Table 2: Results for the MMALA schemes for inference on the FitzHugh–Nagumo model. For each parameter (a,b,c) and algorithm the mean (over the 100 replicates) ESS is presented as well as its standard error. The CPU time and the mean ESS per second for each parameter are also provided.

Method	*ESS (mean)*	*ESS (s.e)*	*CPU Time*	*mean ESS/s*
PMALA	(1639.6, 669.3, 1406.4)	(1.9, 1.2, 1.7)	896.8	(1.83,0.75,1.57)
MMALA	(1274.4, 632.8, 1120.5)	(1.7, 1.2, 1.3)	923.0	(1.38,0.69,1.21)

4.2. Non-linear differential equation model

We now consider the FitzHugh–Nagumo differential equations in [11]: $\dot{W} = c\left(W - W^3/3 + R\right)$ and $\dot{R} = -(W - a + bT)/c$. The simulated dataset and our independent priors for the parameter vector (a, b, c) are the same as those used in [2]. To be consistent with the appendix of [2] and the associated Matlab code we assume $\beta \sim \text{Exp}(1)$.

The mean ESS for each parameter, along with its standard error are shown in Table 2, and it is clear that PMALA outperforms MMALA using this measure. CPU time and ESS/sec are also provided in the table; since each iteration of PMALA is also quicker, its advantage is even clearer when CPU time is accounted for.

Acknowledgements

S. Livingstone is funded by a PhD Scholarship from Xerox Research Centre Europe. S. Byrne is funded by an EPSRC Postdoctoral Research Fellowship, EP/K005723/1. M. Girolami is funded by an EPSRC Established Career Research Fellowship, EP/J016934/1 and a Royal Society Wolfson Research Merit Award.

References

[1] G. O. Roberts, J. S. Rosenthal, Optimal scaling of discrete approximations to langevin diffusions, Journal of the Royal Statistical Society. Series B (Statistical Methodology) 60 (1998) pp. 255–268.

[2] M. Girolami, B. Calderhead, Riemann manifold langevin and hamiltonian monte carlo methods, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73 (2011) 123–214.

[3] G. Roberts, O. Stramer, Langevin diffusions and Metropolis-Hastings algorithms, Methodology And Computing In Applied Probability 4 (2002) 337–357.

[4] R. Durrett, Stochastic Calculus: A Practical Introduction, Probability and Stochastics Series, CRC Press, 1996.

[5] P. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, volume 23 of *Stochastic Modelling and Applied Probability*, Springer, 1992.

[6] B. Oksendal, Stochastic Differential Equations, Springer, 1998.

[7] G. O. Roberts, R. L. Tweedie, Exponential convergence of Langevin distributions and their discrete approximations, Bernoulli 2 (1996) 341–363.
[8] W. Gilks, S. Richardson, D. Spiegelhalter, Markov chain Monte Carlo in Practice, Chapman & Hall/CRC, 1996.
[9] J. Kent, Time-reversible diffusions, Advances in Applied Probability 10 (1978) pp. 819–835.
[10] H. Federer, Geometric measure theory, Die Grundlehren der mathematischen Wissenschaften, Band 153, Springer-Verlag New York Inc., New York, 1969.
[11] J. O. Ramsay, G. Hooker, D. Gampbell, J. Cao, Parameter estimation for differential eequation: a generalized smoothing approach (with discussion), Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69 (2007) 741–796.